ORIGINAL RESEARCH PAPER

Volume - 14 | Issue - 03 | March - 2025 | PRINT ISSN No. 2277 - 8179 | DOI : 10.36106/ijsr

INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH

SELF-SUPERVISED GAIT RECOGNITION WITH DIFFUSION MODEL PRETRAINING

Physiotherapy

Anuj Kabra

BPT, MPT (Orthopedics), Certified McKenzie Therapist, Keller Oaks Healthcare Services, Keller (Texas, USA)

ABSTRACT

Gait recognition has potential as a biometric tool but faces challenges in unconstrained environments due to occlusions and varying conditions. We propose a novel approach using diffusion models for self-supervised pretraining to enhance gait recognition accuracy. The method involves pretraining a gait feature extractor with a conditional latent diffusion model, followed by fine-tuning for specific tasks. Experiments on Gait3D and GREW datasets show significant performance improvements, demonstrating that diffusion model pretraining offers better feature representation and recognition in real-world scenarios. This approach opens new avenues for robust gait analysis.

KEYWORDS

Self supervised gait pretraining, gait analysis,GREW model,Gait3D model, gait feature extractor.

INTRODUCTION

Gait, the unique manner in which a person walks, offers an alternative way to identify individuals, alongside the more common biometric modalities like fingerprints and irises. With the emergence of deep learning, advancements in computer vision architectures, and the collection of well-labelled gait datasets, deep gait recognition has become an increasingly popular area of research over the last few years (Sepas- Moghaddam and Etemad, 2022). Consequently, owing to the efforts of many previous works, it has become possible to achieve impressive accuracy performance on existing controlled datasets such as CASIA-B (Yu et al., 2006) and OU-MVLP (Takemura et al., 2018). However, when these techniques are applied to recently released in-the-wild gait datasets such as Gait3D (Zheng et al., 2022) and GREW (Zhu et al., 2021), their performance pales in comparison, highlighting their limited applicability to unconstrained settings. As a result, recent studies (Zheng et al., 2022; Zhu et al., 2021; Lin et al., 2022; Fan et al., 2023c; Cosma et al., 2023; Fan et al., 2023a; Habib et al., 2024) have begun shifting their focus towards addressing the more challenging problem of gait recognition in the wild. Recently, diffusion models (Sohl- Dickstein et al., 2015; Ho et al., 2020; Song et al., 2020; Nichol and Dhariwal, 2021; Rombach et al., 2022; Song et al., 2023) have been in the spotlight for their impressive performance on generative tasks, outshining variational autoencoders, flow-based generative models, and generative adversarial networks which previously dominated the generative realm. Their stability during training and ability to generate high-quality realistic samples, such as images (Rombach et al., 2022; Nichol et al., 2021; Ramesh et al., 2022; Saharia et al., 2022; Dhariwal and Nichol, 2021), videos (Ho et al., 2022b;a; Singer et al., 2022; Bar-Tal et al., 2024), and audio (Schneider, 2023; Huang et al., 2023; Mittal et al., 2021), have led to their wide adoption in academia and industry. Amidst the hype to exploit diffusion models for generative tasks, other promising directions of diffusion models have been less explored. One such direction involves leveraging diffusion models for representation learning, where a model learns to extract relevant features during the diffusion process, which can be beneficial for tasks beyond generation. Moreover, with a simple reconstruction or noise prediction objective, diffusion training can potentially serve as an effective self-supervised pretraining approach, which does not necessitate the need for a labeled dataset. While some studies have started looking into using the learnt representations for common tasks like image classification (Clark and Jaini, 2024; Hudson et al., 2023; Li et al., 2023; Xiang et al., 2023; Abstreiter et al., 2021; Yang and Wang, 2023) and segmentation (Yang and Wang, 2023; Zhao et al., 2023), not much attention has been paid to using them in applications such as gait recognition. Furthermore, many previous works related to diffusion-based representation learning have not investigated the effects of fine tuning the learnt representations on downstream tasks, overlooking the pretraining potential of diffusion training. Considering the limited research on diffusion-based representation learning in the gait recognition field and the more challenging in-the-wild scenarios, we aim to address the following question: can we leverage diffusion training to enhance existing methods for gait recognition in the wild? To explore this, we propose a diffusion-based approach to pretrain the backbone of a gait recognition model by using its output as a condition for a latent diffusion model. Thereafter, we initialize the gait recognition model with the pretrained backbone and perform transfer learning on the downstream gait recognition task. We conducted extensive applicability studies with multiple existing gait recognition models, including GaitGL (Lin et al., 2022), GaitPart (Fan et al., 2020), GaitSet (Chao et al., 2019), SMPLGait w/o 3D (Zheng et al., 2022), and Gait-Base (Fan et al., 2023c), by evaluating on two in-the-wild gait datasets, Gait3D (Zheng et al., 2022) and GREW (Zhu et al., 2021). Additionally, we performed thorough ablation studies to investigate the effects that different pretraining hyper parameters have on the downstream gait recognition task. Our finding reveals that during diffusion pretraining, the gait recognition model backbone, regardless of its architecture, learns to separate gait sequences belonging to different subjects further apart than those belonging to the same subject, despite the lack of an explicit signal to do so. This results in a steady improvement in the gait recognition performance and demonstrates the potential of diffusion pretraining for gait recognition. Subsequently, when the gait recognition model is initialized with the pretrained backbone and further fine tuned on the downstream gait recognition task, it surpasses the performance of its trained-from-scratch counterpart by as much as 7.9% on Gait3D and 4.2% on GREW. This remains the case even when the number of supervised training iterations is significantly reduced by as much as 89% on Gait3D and 70% on GREW. To the best of our knowledge, we are the first to apply diffusion training for representation learning in the field of gait recognition and demonstrate its effectiveness as a pretraining approach for the gait recognition task.

RELATED WORK.

A. GAIT Recognition

With the advent of deep learning, gait recognition typically involves extracting gait features from gait sequences and projecting these features into more discriminative embeddings that can be compared using a distance metric such as Euclidean or cosine distances. These gait sequences typically come in the form of either silhouettes or skeletons, with silhouette- based recognition being more popular (Sepas-Moghaddam and Etemad, 2022), though skeleton-based gait recognition remains an active area of research (Teepe et al., 2021; 2022; Zhang et al., 2023). Much state-of-the-art research regarding deep gait recognition focuses on the architectural design of backbone networks to maximize the extraction of meaningful gait information from gait sequences (Lin et al., 2022; Fan et al., 2023c; Cosma et al., 2023; Fan et al., 2023a; Chao et al., 2019; Fan et al., 2020; Song et al., 2019). For instance, GaitPart (Fan et al., 2020) proposes a part-based approach that divides each gait silhouette into several parts and extracts features for each part separately to obtain more fine-grained features. It also focuses on short-range neighboring frames rather than considering all frames within a gait sequence. GaitSet (Chao et al., 2019), on the other hand, treats gait as an unordered set of gait silhouettes and uses only permutation-invariant operations within its architecture. Inspired by the fields of person reidentification, current state-of-the-art research (Chao et al., 2019; Fan et al., 2020; Lin et al., 2022; Zheng et al., 2022; Fan et al., 2023c) also adapted techniques such as horizontal pyramid matching (Fu et al., 2019) and batch normalization neck (Luo et al., 2019) to further enhance gait recognition accuracy. Several works have also explored self-supervised learning approaches (Liu et al., 2021; Rao et al., 2021; Fan et al., 2023b; Cosma et al., 2023). They are often based on contrastive learning, where augmented versions of the same gait sequences are treated as positive pairs and different gait sequences are treated as negative pairs. While many of these methods have performed well on existing controlled datasets such as CASIA-B (Yu et al., 2006) and OU-MVLP (Takemura et al., 2018), their performance drastically drops when applied to the recently released in-the-wild datasets (Zheng et al., 2022; Zhu et al., 2021), where factors such as temporary occlusions, varying camera viewpoints, and illumination inadvertently come into play. Finding a way that can universally enhance the performance of these methods in such unconstrained settings is highly desirable. As the introduction of in-the-wild gait datasets is fairly recent, many earlier works did not have the opportunity to evaluate their methods on these datasets. However, with the efforts of OpenGait (Fan et al., 2023c), several previous works have been reproduced, trained, and evaluated on the in-the-wild datasets, serving as baselines for future research in gait recognition in the wild.

B.Diffusion~ N First proposed by Sohl-Dickstein et al. (2015), diffusion models, particularly Denoising Diffusion Probabilistic Models (Ho et al., 2020), are now an active area of research. A typical diffusion process consists of two phases—the forward phase and the reverse phase. During the forward phase, a data sample *x* is gradually transformed into an approximate pure noise by iteratively adding noise ϵ (0, *I*) to a data sample across a series of timesteps based on a defined noise schedule α^{-}_{t}. At any timestep *t* of the forward diffusion process, the noised data sample x_t can be expressed as:

$$x_t = \sqrt{\alpha^{-}_{t}} \cdot x + \sqrt{1 - \alpha^{-}_{t}} \cdot \epsilon \qquad (1)$$

On the other hand, during the reverse phase, noise is iteratively removed from a noisy sample until a clean sample is obtained. By training a model ϵ_θ to predict the added noise ϵ in the noisy data sample x_t at any given timestep *t* of the forward diffusion process, a mapping from random noise to the data manifold can be learnt, which confers diffusion models their generative ability. The overall training objective of diffusion models can be summarized as the minimization of the following noise prediction loss:

$$L = \epsilon - \epsilon_\theta(x_t, t)\,^2 (2)$$

In order to generate more precise data samples, diffusion models are fed with an additional prompt and instead learn a conditional data distribution. In this case, the above loss becomes:

$$L = \epsilon - \epsilon_\theta(x_t, t, c)\,^2 \qquad (3)$$

Various methods have been proposed to improve the quality of samples produced by conditional diffusion models. In particular, classifier-free guidance which randomly drops out the added condition, has been one of the most simple and effective. To reduce the computation cost and memory consumption brought about by training with and generating high-resolution data, Rombach et al. (2022) proposed latent diffusion, whereby data are first encoded into a lower-dimensional latent space before the diffusion process is applied. To further optimize the training efficiency of diffusion models, various noise schedulers and timestep weighting approaches have also been proposed. These methods serve to prioritize a certain noise range during the diffusion process, which improves convergence.

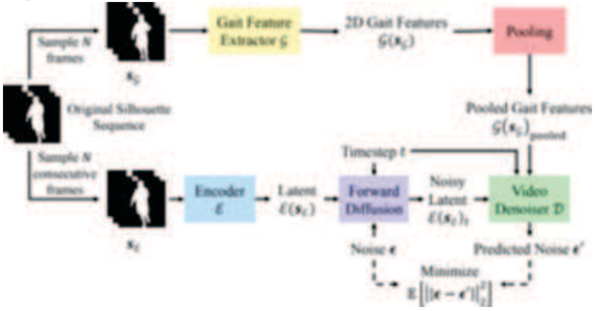
Sample N frames
x_G
Gait Feature Extractor $\mathcal{G}$
2D Gait Features $\mathcal{G}(x_G)$
Pooling
Original Silhouette Sequence
Pooled Gait Features $\mathcal{G}(x_G)_{pooled}$
Sample N consecutive frames
Timestep t
Encoder $\mathcal{E}$
Latent $\mathcal{E}(x_E)$
Forward Diffusion
Noisy Latent $\mathcal{E}(x_E)_t$
Video Denoiser $\mathcal{D}$
x_E
Noise ϵ
Predicted Noise ϵ'
Minimize
$\mathbb{E}[\|\epsilon - \epsilon'\|_2^2]$

Fig. 1. Proposed architecture. Only the gait feature extractor and video denoiser are trained during diffusion pretraining

Initially started on image generation diffusion models have now evolved to generate videos audio and even architecture- specific neural network parameters. Despite its impressive generative feats in various domains, the learnt representations produced by the diffusion process have not been looked into as intensively, particularly in specialized fields such as gait recognition. Closely related to our work, Hudson et al. (2023) assessed the usefulness of the diffusion process for gait recognition representations learnt by training an encoder and a denoiser with only diffusion and evaluating them through linear-probing experiments on image classification tasks. However, the usefulness of the learnt representations in enhancing the performance on downstream tasks has not been looked into.

III. PROPOSED METHOD.

In this section, we introduce our proposed method, which consists of two stages:

(1) pretraining with diffusion and (2) transfer learning on the downstream gait recognition task.

A. Diffusion Pretraining

The encoder *E* transforms the input silhouette sequence into a compact latent representation. This process is essential for reducing the dimensionality of the input while retaining relevant information for subsequent processing. The denoiser *D* is responsible for removing the noise added to the latent representation during the diffusion process. It utilizes the features provided by the gait feature extractor *G* to improve the precision of the denoising process. The gait feature extractor *G* identifies key features from the input silhouette sequence, which are then passed on as a condition to the denoiser. This helps the denoiser focus on the relevant aspects of the sequence, leading to a more accurate reconstruction.

1)Encoder: Considering its small size and decent silhouette-encoding ability, we use the open- sourced Tiny AutoEncoder for Stable Diffusion (TAESD) as our encoder. As silhouette sequences are made up of single- channel grayscale frames and TAESD only accepts three-channel images, silhouette sequences are first replicated along the channel dimension before encoding. Given a three-channel sequence s of shape (t, 3, h, w), where t, h, and w denote sequence length, height, and width, respectively, the encoder of TAESD, E, performs framewise encoding of the sequence along the spatial dimension while preserving the temporal dimension, producing a 48× compressed latent encoding *E*(*s*) of shape (t, 4,).

2) Denoiser: As for the denoiser, given the lack of open- sourced pretrained gait silhouette sequence diffusion models, we adapt a recent video diffusion model architecture proposed by Ho et al. (2022b) to fit our use case. The denoiser accepts noisy latent representations, selected diffusion timesteps, and any other one-dimensional conditions as inputs, and outputs a prediction of the added noise.

3) GAIT Feature Extractor: The gait feature extractor is our target model to be trained by the diffusion process. To aid the denoiser in predicting the added noise, an additional input condition, corresponding to the gait features of the encoded input sequence into the denoiser, is further provided to the denoiser. Given a silhouette sequence s, the gait feature extractor G transforms it into a two-dimensional gait feature, G(s), which is representative of the gait contained within the sequence. The gait feature is extracted by the backbone of a typical deep gait recognition model, which can be based on any previous work. To test the applicability of our approach, we adopt a variety of existing backbones in literature to be the gait feature extractor—GaitGL (Lin et al., 2022), GaitPart (Fan et al., 2020), GaitSet (Chao et al., 2019), SMPLGait w/o 3D (Zheng et al., 2022), and GaitBase (Fan et al., 2023c).

4)Poooling Method: As the denoiser accepts only one- dimensional tensors as conditions while the extracted gait features are two-dimensional tensors, an operation is needed to convert the gait feature tensors into a single dimension. A pooling operation is applied, rather than simply flattening, to reduce the computation that will be introduced from conditioning. In particular, mean pooling is used, following our ablation study in Sec. 5. The pooled gait feature conditions, G(s)pooled, are then concatenated with the timestep conditions and are used to scale and bias the activations within the denoiser layers, following the work of Dhariwal and Nichol (2021).

5)Input Pretreatment: To allow the gait extractor to focus learning on gait information rather than video information, the autoencoder and gait feature extractor are presented with different subsequences sampled from the same silhouette sequence via different sampling algorithms. We denote this pair of input subsequences as $s = (s_E, s_G)$, where s_E denotes an input subsequence to the encoder and s_G denotes an input subsequence to the gait feature extractor, with $s_E = s_G$. As a video diffusion model is used for the denoiser, its input needs to be temporally consistent. Therefore, N frames are sampled consecutively from the silhouette sequence to serve as s_E. As for s_G, we sample N frames using the exact sampling algorithm employed by the authors who proposed the respective gait feature extractor architecture .We fixed $N = 30$ throughout the study. Even though s_E and s_G are different, $G(s_G)$ should ideally correspond to the gait features of s_E, since s_E and s_G belong to the same subject. Each frame in s_G is first normalized and resized to 64 × 64 before being cropped to 64 × 44, which is the common input size for many gait recognition models. As for s_E, they are simply normalized and resized to 64 × 64. To increase the generalizability of the model, s_E and s_G are augmented separately with a combination of RandomAffine, RandomPerspective, RandomHorizontalFlip, RandomPartDilate and RandomPartBlur provides a visualization of each augmentation applied.

6)Noise Scheduler and Loss Weighing Strategy: For generative purposes, noise schedulers, such as the cosine noise scheduler, which prioritize high and low noise levels are commonly adopted. However, in the case of representation learning, Hudson et al. (2023) have found that prioritizing medium noise is more helpful. Following their work, we adopt an inverted cosine noise scheduler. To further focus on medium-level noise, we adopt a medium noise prioritization weighting strategy that down weighs the loss when the noise is too high or too low. Specifically, we modify the existing Min-SNR weighting strategy (Hang et al., 2023), which down weighs losses from low-level noise, to also down weigh losses from high levels of noise.

7)Loss Function: To pretrain our gait recognition model via diffusion, we employ the L2 noise prediction loss commonly used by diffusion models, which can be summarized as:

$$L_{\text{diffusion}} = \left\| \epsilon - DE(s_E)^t, t, G(s_G)_{\text{pooled}} \right\|^2 \qquad (4)$$

Where ϵ is the noise added to the input, $E(s_E)^t$ denotes the encoded subsequence from the encoder at time t, and $G(sG^{)}$pooled represents the pooled features from the gait feature extractor. $E(s_E)^t$ is the noised latent representation fed to the denoiser D, at timestep t (0, 1000] of the forward diffusion process, $\epsilon \sim \mathbf{N}(0, I)$ is the random noise added, and $G(s_G)_{\text{pooled}}$ is the pooled gait feature condition. During diffusion pretraining, the denoiser and gait feature extractor are trained from scratch, while the pretrained TAESD encoder is kept frozen. A higher learning rate is assigned to the gait feature extractor to enable it to better guide the denoiser during training, following Hudson et al. (2023).

B. Transfer Learning

Once the gait feature extractor is trained by diffusion, we evaluate it on the downstream gait recognition task. We replicate the remaining parts of the gait recognition model accordingly and initialize the weights of the gait backbone with the ones learned during diffusion pretraining. The untrained parameters are initialized based on the settings provided by OpenGait. We consider two cases of transfer learning, namely, with and without fine-tuning of the pretrained backbone. During the first case, the pretrained backbone is frozen to evaluate the usefulness of the learned representations for the downstream gait recognition task. In the latter case, the pre- trained backbone is allowed to be fine-tuned together with the untrained parameters to evaluate the effectiveness of diffusion as a pretraining method. During transfer learning, each gait recognition model is trained using the standard triplet loss, L_{triplet}, for identification. The distance measure typically employed is the Euclidean distance. However, we employ the cosine distance instead, as we found that the use of cosine distance during training and evaluation produces much better results for existing works Additionally, some methods, such as GaitBase and SMPL- Gait w/o 3D, also incorporate an additional smoothed identity loss, L_{ID}, by having another module predict the identity of each gait sequence to enhance the gait recognition performance. In this case, the net loss is the sum of the triplet loss and the reweighted smoothed identity loss:

$$L_{\text{net}} = L_{\text{triplet}} + 0.1 \cdot L_{\text{ID}} \qquad (5)$$

IV. Experiments

A. Experimental Setup

With the focus on practical gait recognition, two datasets meant for gait recognition in the wild, namely, Gait3D and GREW, were chosen. Note that only silhouette sequences from these datasets were used. For pretraining, we used AdamW with warmup and cosine annealing with learning rate of $1{\times}10^{-4}$ for the denoiser and $5{\times}10^{-4}$ for the gait feature extractor. The models were trained for 120k iterations with a batch size of 64 for Gait3D and 128 for GREW. For transfer learning, we trained and evaluated on the same dataset that was used during diffusion pretraining. We used the rank-1 gait recognition accuracy as our main evaluation metric. All transfer learning hyperparameters, unless mentioned in Sec. 4.3, were kept the same as those provided in the OpenGait framework. For GREW, we submitted the results to the official website for evaluation. To evaluate the effectiveness of our proposed method, we reproduced the results of the corresponding gait recognition models on the Gait3D and GREW datasets trained solely via the supervised objective using the OpenGait framework. Recognizing that data augmentation can enhance the performance of gait recognition models, we also included the case when it is applied. The performance of each reproduced baseline is presented in Table I.

TABLE I RANK-1 ACCURACY ON GAIT3D AND GREW. GAITPART ON GREW IS EXCLUDED DUE TO INSTABILITY DURING TRAINING WITH COSINE DISTANCE. FOR TRAIN ITERATION, X + Y DENOTE XK ITERATIONS OF DIFFUSION PRETRAINING FOLLOWED BY YK ITERATIONS OF TRANSFER LEARNING. TRANSFER LEARNING ITERATIONS FOR GAITBASE ON GREW WITHOUT AND WITH DATA AUGMENTATION ARE 90K AND 120K, RESPECTIVELY.

Method	**Gait3D Rank-1 Accuracy (%)**	**Train Iter.**	**GREW Rank-1 Accuracy (%)**	**Train Iter.**		
	×Data Aug.	Data Aug. (×*103*)	×Data Aug.	Data Aug. (×*103*)		
Reproduced Baseline						
GaitGL	29.2	32.4	54.0	58.4	180	250
GaitPart	31.2	38.7	-	-	180	-
GaitSet	42.2	47.8	48.1	53.1	180	250
SMPLGait w/o 3D	45.5	42.9	47.6	52.1	180	250
GaitBase	56.5	65.8	58.1	61.8	60	180
Diffusion Pretraining + Transfer Learning with Frozen Back- bone						
GaitGL	17.0	34.4	32.2	56.3	120 + 60	120 + 125
GaitPart	18.8	35.7	-	-	120 + 60	-
GaitSet	23.5	33.5	33.5	52.0	120 + 60	120 + 125
SMPLGait w/o 3D	30.7	36.0	36.0	51.8	120 + 60	120 + 125
GaitBase	35.0	40.5	40.5	58.5	120 + 60	120 + 90
Diffusion Pretraining + Transfer Learning with Finetuning of Backbone						
GaitGL	34.4 (↑ 5.2)	34.4 (↑ 2.0)	56.3 (↑ 2.3)	58.6 (↑ 0.2)	120 + 60	120 + 125
GaitPart	35.7 (↑ 4.5)	41.7 (↑ 3.0)	-	-	120 + 60	-
GaitSet	45.0 (↑ 2.8)	49.9 (↑ 2.1)	52.0 (↑ 3.9)	55.4 (↑ 2.3)	120 + 60	120 + 125
SMPLGait w/o 3D	53.4 (↑ 7.9)	60.7(↑17.8)	51.8 (↑ 4.2)	54.1 (↑ 2.0)	120 + 60	120 + 125
GaitBase	62.3 (↑ 5.8)	69.7 (↑ 3.9)	58.5 (↑ 0.4)	62.0 (↑ 0.2)	120 + 60	120 + 90/120

B. Pretraining Results

To examine if the gait feature extractor has learnt something useful for gait recognition during diffusion pretraining, we measured the gait recognition performance at different check- points of the pretraining process (Fig. 2). This was done by feeding the test set's silhouette sequences into the gait feature extractor and directly using either the cosine or Euclidean distance between the output gait features to determine their similarity. For convenience, the accuracy observed during pretraining for GREW was based on the test set defined in OpenGait. Due to space constraints, we only show the findings for one of the gait feature extractors, GaitSet, as it was pretrained on Gait3D and GREW. Interestingly, we observed a steady improvement in gait recognition performance during the diffusion pretraining process, showing that the gait extractor had undoubtedly learnt to extract some features useful for gait recognition. Note that we are simply reconstructing the inputs at this stage.

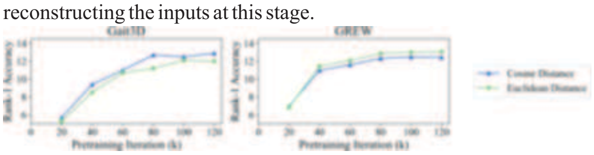

Fig. 2. Rank-1 accuracy curves during diffusion pretraining on Gait3D and GREW (GaitSet).

To investigate further, we recorded the mean cosine distance of the gait features of anchor-positive pairs and the gait features of anchor-negative pairs within a batch during pretraining on Gait3D and GREW (Fig. 3). In our case, an anchor-positive pair refers to an augmented pair (s_E, s_G) sampled from the same silhouette sequence. In contrast, an anchor-negative pair refers to an augmented pair found within the batch where s_E and s_G are sampled from different silhouette sequences.

Looking at Fig.3,we observe an interesting phenomenon—the difference in the cosine distances between the anchor-positive pairs and anchor-negative pairs increases and stabilizes during diffusion pretraining. This is independent of the architecture of the gait feature extractor as well as the dataset, suggesting that our proposed diffusion pretraining approach trains the gait feature extractor to maintain some margin of separation between the anchor-positive pairs and anchor-negative pairs. This is in spite of not having any supervisory signal to encourage separation during training and likely explains why we see an improvement in gait recognition performance even at this stage.

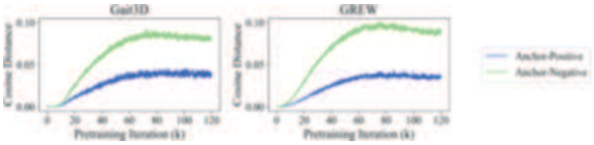

Fig. 3. Mean cosine distance of anchor-positive pair and anchor-negative pair during diffusion pretraining on Gait3D and GREW (GaitSet)

C. Transfer Learning Results

1)Frozen Backbone: With diffusion pretraining, we have pretrained the backbones of various existing gait recognition models. What is left is to project the features into more discriminative embeddings with the remaining parts of these models for the downstream gait recognition task. To evaluate how discriminative the learnt features are for gait recognition, we froze the pretrained backbones and simply trained the remaining untrained layers via the supervised learning objective. The results are shown in Table 1. Through transfer learning, we achieved higher gait recognition accuracy compared to using the gait features alone. That said, compared to what could be achieved solely by supervised learning on the gait recognition task, the learnt gait features are not as discriminative. This is likely the case since not all the features learnt for input reconstruction are relevant to gait recognition. Nonetheless, some discriminative features are learnt during diffusion training, highlighting its pretraining potential.

2)Finetuning of the Backbone: Next, we allowed the pretrained backbones to be finetuned to investigate if improvements could be made. During finetuning, we found that the ratio of the learning rate of the pretrained layers to that of the untrained layers, denoted as *r*, is an important hyperparameter. We attempted *r* for both Gait3D and GREW datasets. Moreover, for the GREW dataset, different from the baseline settings with zero weight decay, we added a small weight decay term of 5×10^{-5} during the fine tuning of GaitGL, GaitSet, and SMPLGait w/o 3D, as we observed cases of over fitting with the pretrained backbone. Table 1 shows the best finetuning results obtained with the corresponding value of *r* used. Looking at Table 1, we observe that all models initialized with the diffusion-pretrained backbone exhibit improved gait recognition performance compared to their trained-from-scratch counterparts, even with a significantly reduced number of supervised training iterations. Excluding the anomalous case for SMPLGait w/o 3D which deteriorated with data augmentation on Gait3D, we observe improvements in rank-1 accuracy by as much as 7.9% and 4.2% on the Gait3D and GREW datasets, respectively. For SMPLGait w/o 3D, which performed poorly with data augmentation on the Gait3D dataset when trained from scratch, initializing its backbone with diffusion-pretrained weights helped to overcome its poor initialization. This underscores the effectiveness of diffusion pretraining as a method to provide a decent initialization point for the downstream gait recognition task.

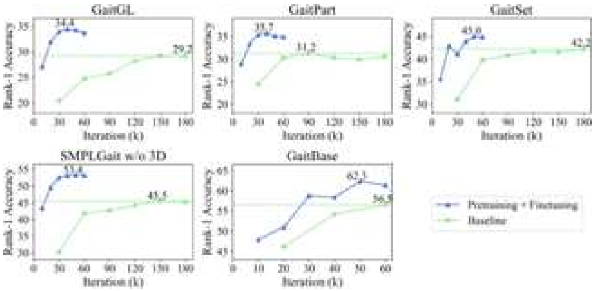

Fig. 4. Rank-1 gait recognition accuracy curves when no data augmentation is applied during supervised training (Gait3D).

To further prove our point, we show the rank-1 accuracy curves of the finetuned models with their respective baselines on the Gait3D dataset for the case when no data augmentation is applied during supervised training (Fig. 4). With the backbone pretrained by diffusion, the models outperformed their corresponding supervised baselines within as little as 20k iterations, corresponding to an 89% reduction in supervised training iterations. As for GREW, we show that GaitGL, GaitSet, and SMPLGait w/o 3D were still able to achieve competitive performance when the supervised training iterations were further reduced to 30% of their baselines' total training iteration (Table II).

Table II Rank-1 Accuracy On Grew With Further Reduction In Supervised Training Iterations

Method	GREW Rank-1 Accuracy (%)	Train Iter.	
	×Data Aug.	✓ Data Aug. (×*103*)	
GaitGL	54.3	56.1	120 + 75
GaitSet	51.1	53.3	120 + 75
SMPLGait w/o 3D	50.3	51.8	120 + 75

Ablation Studies:

In this section, we present the various ablation studies conducted to investigate the effects that different hyperparameter settings have on the diffusion pretraining process and downstream gait recognition tasks. For all experiments, we used the backbone of SMPLGait w/o 3D as our gait feature extractor and Gait3D as the dataset. During transfer learning, no data augmentation was applied, and the pretrained backbone was finetuned with a lower learning rate(0.1)than the untrained layers. Aside from the hyperparameter being investigated, all other hyperparameters were assigned based on the default pretraining and transfer learning settings. The results of the various ablation studies are summarized in Table III.

Noise Scheduler and Loss Weighting: We explored two kinds of schedulers—a typical cosine scheduler and an inverted cosine scheduler. For the cosine scheduler, we attempted a uniform weighting strategy, where losses from different timesteps are weighed equally, and the Min-SNR strategy [15]. As for the inverted cosine scheduler, we attempted uniform weighting and our proposed medium noise prioritization (MNP) weighting strategy, which downweighs losses from both low and high timesteps. We found that focusing on medium-level noise through the combined use of an inverted cosine scheduler and the MNP weighting strategy worked the best.

Feature Pooling Method: We investigated different methods to pool the two-dimensional output of the gait feature extractor into a one-dimensional condition—mean pooling, max pooling, and the sum of the mean and max. Out of the three pooling methods, mean pooling worked the best

TABLE III Downstream Rank-1 Gait Recognition Accuracy On Gait3d For Different Diffusion Pretraining Hyperparameter Settings. The First Row Shows The Default Setting We Used For Gait3d. Highlighted Entries Denote The Different Hyperparameter Settings Compared To Our Default Setting.

Noise Scheduler	Loss Weighting	Pooling	Data Augmentation	Denoiser Size	lG/lD	Puncond	R-1 (%)
Inverted Cosine	MNP	Mean	√	11.6 M	5	0.15	49.1
Inverted Cosine	Uniform	Mean	√	11.6 M	5	0.15	47.6

Cosine	Uniform	Mean	√	11.6 M	5	0.15	47.5
Cosine	Min-SNR	Mean	√	11.6 M	5	0.15	48.2
Inverted Cosine	MNP	Max	√	11.6 M	5	0.15	47.8
Inverted Cosine	MNP	Mean + Max	√	11.6 M	5	0.15	47.4
Inverted Cosine	MNP	Mean		11.6 M	5	0.15	44.6
Inverted Cosine	MNP	Mean	√	3.7 M	5	0.15	47.5
Inverted Cosine	MNP	Mean	√	40.2 M	5	0.15	50.6
Inverted Cosine	MNP	Mean	√	11.6 M	1	0.15	45.9
Inverted Cosine	MNP	Mean	√	11.6 M	2	0.15	47.4
Inverted Cosine	MNP	Mean	√	11.6 M	5	0.00	47.2
Inverted Cosine	MNP	Mean	√	11.6 M	5	0.50	47.7

Inputs to Denoiser and Gait Feature Extractor: We investigated what would happen during diffusion pretraining if the input to the gait extractor s_G and the input to the autoencoder s_E were identical. Fig. 5 shows the rank-1 gait recognition accuracy curves of the gait feature extractor during the diffusion pretraining process when $s_E \neq s_G$ and $s_E = s_G$. When $s_E \neq s_G$, we observe a steady improvement in gait recognition accuracy during diffusion pretraining. However, when $s_E = s_G$, the gait recognition accuracy fluctuates over time, suggesting that what the gait feature extractor is learning during diffusion pretraining is unlikely to be effective gait features. Instead, it could be extracting video information to aid the denoiser in the denoising task.

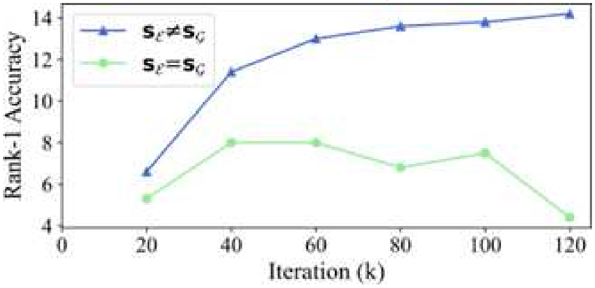

Fig. 5. Rank-1 accuracy curves during diffusion pretraining on Gait3D when $s_E \neq s_G$ and $s_E = s_G$ (SMPLGait w/o 3D).

Data Augmentation During Pretraining:

We turned off data augmentation during diffusion pretraining to investigate its impact on downstream gait recognition performance. Without any data augmentation during pretraining, a significant drop in the downstream gait recognition performance was observed, highlighting the necessity of data augmentation during diffusion pretraining.

Size of Denoiser:

We varied the size of the denoiser used during the diffusion pretraining process by changing its initial channel dimension. We found that the larger the denoiser, the better the downstream gait recognition performance. This suggests that further improvements can be made to the downstream task by increasing the denoiser size during diffusion pretraining, although it would come at the cost of larger memory consumption and longer pretraining time.

Learning Rate Of Denoiser And Gait Feature Extractor:

We investigated how different relative ratios of the learning rate between the gait feature extractor and denoiser, affect downstream performance. We kept the denoiser learning rate constant at 1×10^{-4} and varied the learning rate of the gait extractor. We observed that increasing led to better downstream performance.

Unconditional Training Probability:

To investigate the effects of varying the unconditional training probabil- ity, P_{uncond}, during diffusion pretraining, we explored P_{uncond} We found that our proposed diffusion pretraining approach benefits from classifier free guidance. However, a high unconditional training probability ended up hurting downstream gait recognition performance, likely because the gait feature extractor was updated less. A moderate unconditional training probability yielded the best result.

CONCLUSION:

In summary, we introduce a diffusion pretraining approach for gait recognition in the wild. By simply conditioning a diffusion denoiser with the output of a gait feature extractor, we can pretrain the gait feature extractor to extract relevant features for gait recognition. Initializing the gait recognition model with the pretrained backbone and training it on the downstream gait recognition task further allows us to surpass the performance of its supervised learning counterpart within a much shorter supervised training duration. Our experiments on the Gait3D and GREW datasets demonstrated the broad applicability of our method, achieving rank-1 gait recognition accuracy improvements of up to 7.9% for Gait3D and 4.2% for GREW. We hope our work will spur further interest among researchers in employing diffusion models for representation learning, not only in the gait recognition field but also in other fields as well.

REFERENCES:

[1] Korbinian Abstreiter, Sarthak Mittal, Stefan Bauer, Bernhard Scholkopf, and Arash Mehrjou, "Diffusion-based representation learning," *arXiv preprint arXiv:2105.14257*, 2021.

[2] Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss Shiran Zada, Ariel Ephrat, Junhwa Hur, Yuanzhen Li, Tomer Michaeli, et al., "Lumiere: A space-time diffusion model for video generation," *arXiv preprint arXiv:2401.12945*, 2024.

[3] Ollin Boer Bohan, "Tiny autoencoder for stable diffusion," https://github.com/madebyollin/taesd, 2023.

[4] Hanqing Chao, Yiwei He, Junping Zhang, and Jianfeng Feng, "Gaitset: Regarding gait as a set for cross-view gait recognition," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 8126–8133, 2019.

[5] Kevin Clark and Priyank Jaini, "Text-to-image diffusion models are zero shot classifiers," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[6] Adrian Cosma, Andy Catruna, and Emilian Radoi, "Exploring self supervised vision transformers for gait recognition in the wild," *Sensors*, vol. 23, no. 5, pp. 2680, 2023.

[7] Prafulla Dhariwal and Alexander Nichol, "Diffusion models beat GANs on image synthesis," *Advances in Neural Information Processing Systems*, vol. 34, pp. 8780–8794, 2021.

[8] Chao Fan, Yunjie Peng, Chunshui Cao, Xu Liu, Saihui Hou, Jiannan Chi, Yongzhen Huang, Qing Li, and Zhiqiang He, "Gaitpart: Temporal part- based model for gait recognition," in *Proceedings of the IEEE/CVF con- ference on computer vision and pattern recognition*, pp. 14225–14233, 2020

[9] Chao Fan, Saihui Hou, Yongzhen Huang, and Shiqi Yu, "Exploring deep models for practical gait recognition," *arXiv preprint arXiv:2303.03301*, 2023a.

[10] Chao Fan, Saihui Hou, Jilong Wang, Yongzhen Huang, and Shiqi Yu" Learning gait representation from massive unlabelled walking videos: A benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023b.

[11] Chao Fan, Junhao Liang, Chuanfu Shen, Saihui Hou, Yongzhen Huang and Shiqi Yu, "Opengait: Revisiting gait recognition towards better practicality," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9707–9716, 2023c.

[12] Yang Fu, Yunchao Wei, Yuqian Zhou, Honghui Shi, Gao Huang Xinchao Wang, Zhiqiang Yao, and Thomas Huang, "Horizontal pyramid matching for person reidentification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, pp. 8295–8302, 2019.

[13] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feed forward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 249–256. JMLR Workshop and Conference Proceedings, 2010.

[14] Gavriel Habib, Noa Barzilay, Or Shimshi, Rami Ben-Ari, and Nir Darshan, "Watch where you head: A view-biased domain gap in gait recognition and unsupervised adaptation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6109–6119, 2024

[15] Tiankai Hang, Shuyang Gu, Chen Li, Jianmin Bao, Dong Chen, Han Hu, Xin Geng, and Baining Guo, "Efficient diffusion training via min SNR weighting strategy," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7441–7451, 2023.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.

[17] Jonathan Ho and Tim Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022

[18] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," *Advances in Neural Information Processing Systems* vol. 33, pp. 6840–6851, 2020.